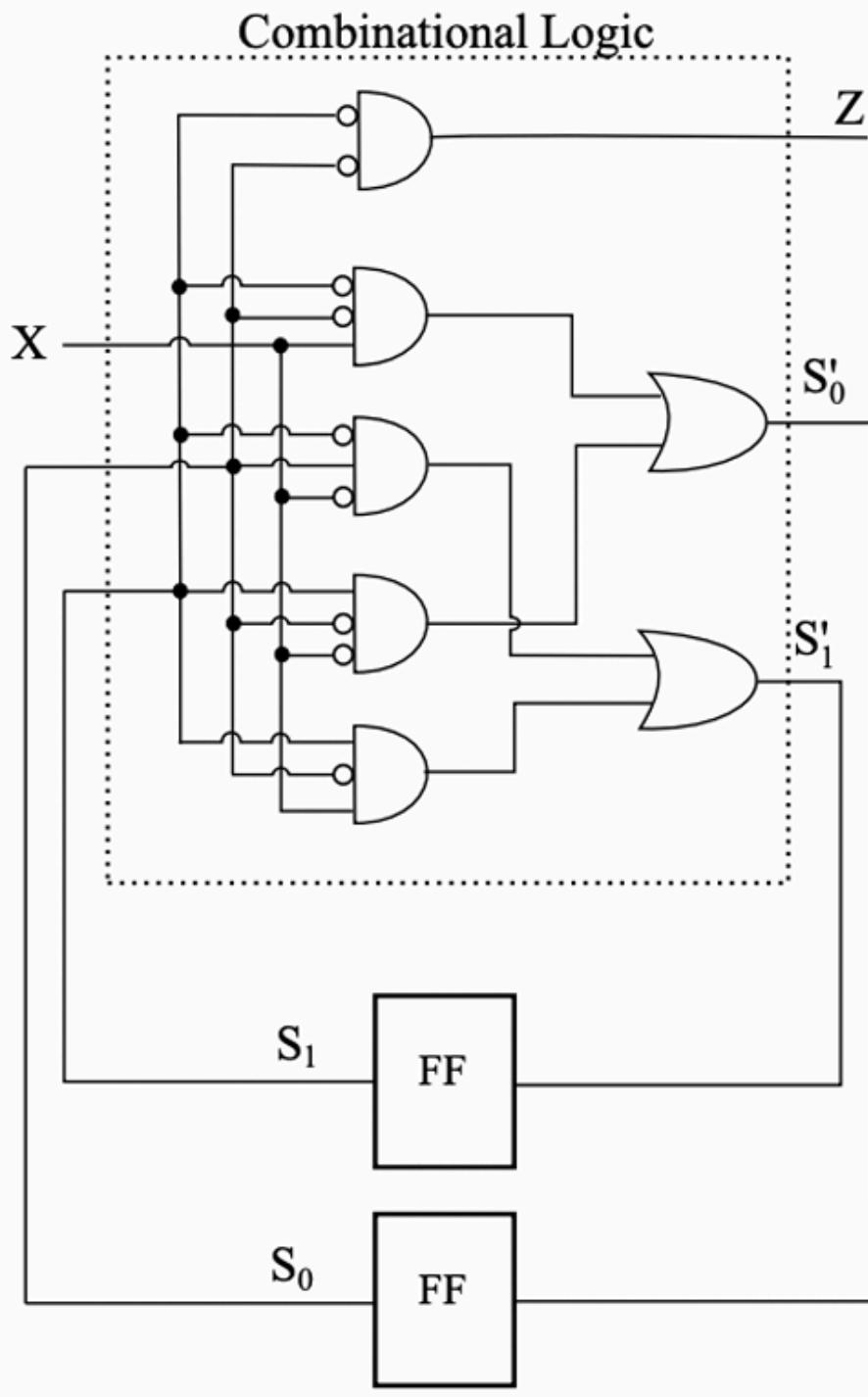


# T1

下图所示的逻辑图是一个有限状态机。



(1). 构造组合逻辑的真值表

$S_1$	$S_0$	$X$	$Z$	$S'_1$	$S'_0$
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

(2). 画出状态机。

## T2

一个LC-3程序存储在内存地址x3000到x3005中。注意，内存地址x3002中的分支指令有一个未指定的PCoffset9，记为X。

Address	Instruction
x3000	0101 000 000 1 00000
x3001	0001 000 000 1 00010
x3002	0000 011 X
x3003	0001 000 000 1 00011
x3004	0001 000 000 1 00001
x3005	1111 0000 0010 0101

程序从PC = x4000开始执行。

你的任务：在下表中，对于X的每个值，回答问题：“程序是否会停止？”（是或否）。如果你的答案是“是”，回答问题：“在x4004处的指令执行完成后，R0中立即存储的值是多少？”如果你的答案是“否”，在“R0中存储的值”列中填入一个破折号。

X	Does the program halt?	Value stored in R0
000000010		
000000001		
000000000		
111111111		
111111110		

## T3

假设从内存读取或写入需要100个周期，而从寄存器读取或写入只需要一个周期。

计算LC-3指令的每个指令周期阶段所需的周期数

1 | ADD R6, R2, R6

假设每个阶段（如果需要）需要一个周期，除非需要访问内存。

## T4

假设我们将LC-3改为只有4个寄存器而不是8个。更少的寄存器通常是一个坏主意，因为这意味着需要更频繁地从内存加载和存储到内存，但我们仍然可以问这个问题：减少寄存器数量会有什么好处吗？对于以下每个问题，回答是或否，并解释你的答案。

1. 如果我们保持所有指令的基本格式不变（并保持每条指令为16位），那么如果我们将寄存器数量减少到4个，对于操作指令（0001, 0101, 1001）是否有任何好处？
2. 如果我们将寄存器数量减少到4个，对于加载（0010）和存储（0011）指令是否有任何好处？
3. 如果我们将寄存器数量减少到4个，对于条件分支（0000）指令是否有任何好处？

## T5

假设一条32位指令采用以下格式：

OPCODE	DR	SR1	SR2	UNUSED
--------	----	-----	-----	--------

如果有225个操作码和120个寄存器，

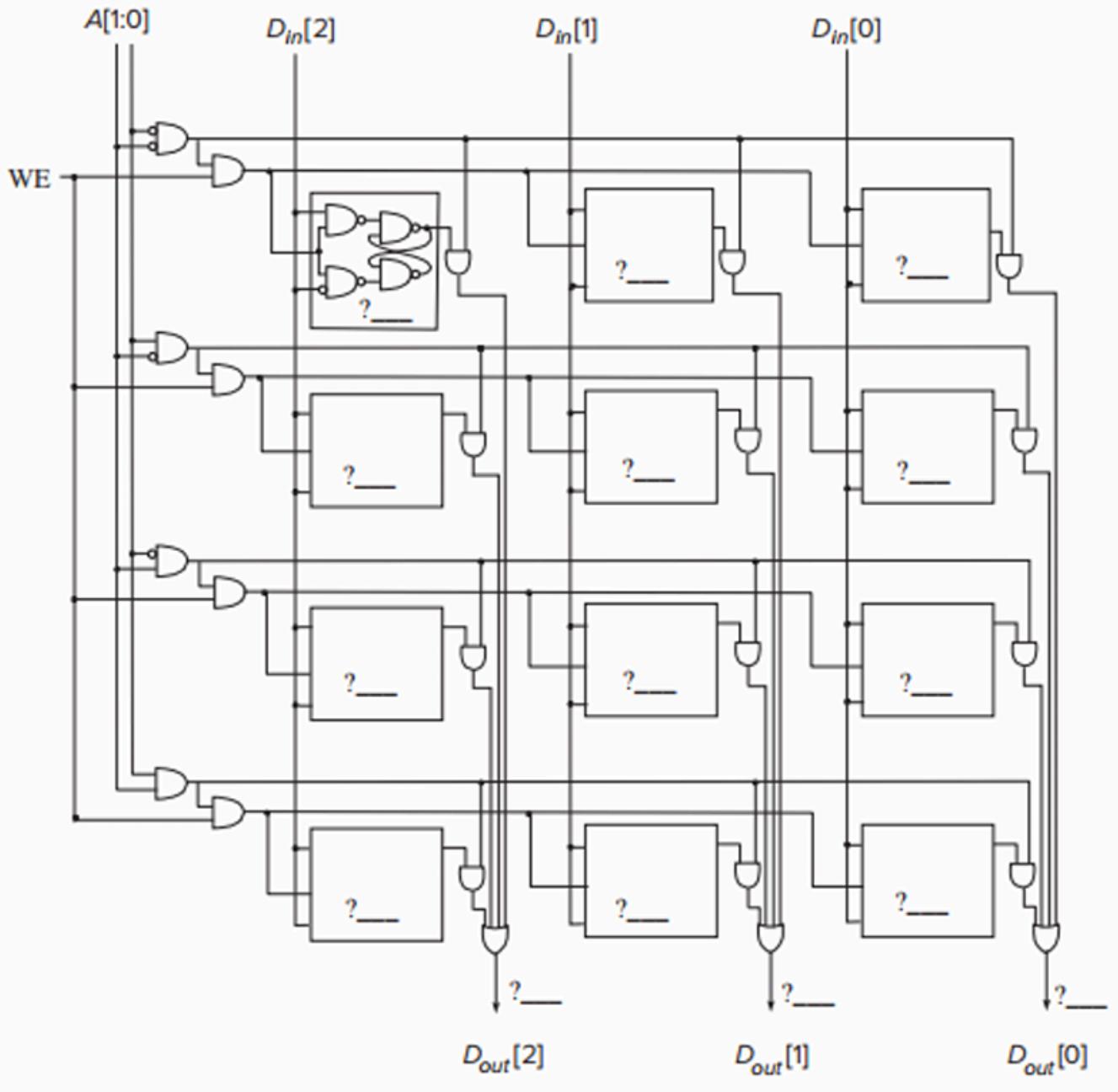
- (a). 表示操作码 (OPCODE) 所需的最小位数是多少?
- (b). 表示目标寄存器 (DR) 所需的最小位数是多少?
- (c). 指令编码中未使用位的最大数量是多少?

## T6

课上讨论的 $2^2 \times 3$ 位内存在连续的五个时钟周期内被访问。下表显示了每次访问期间两位地址、一位写使能和三位数据输入信号的值。

	A[1:0]	WE	D <sub>in</sub> [2:0]
cycle 1	0 1	1	1 0 1
cycle 2	1 1	0	1 1 0
cycle 3	1 0	1	0 1 0
cycle 4	0 1	1	0 1 1
cycle 5	1 1	0	1 0 1
cycle 6	0 0	1	1 1 1
cycle 7	1 1	1	1 1 1
cycle 8	1 1	0	0 1 0

你的任务：填写第八个周期结束前每个内存单元中存储的值和三条数据输出线的值。假设最初所有12个内存单元都存储值1。在下图中，每个问号 (?) 表示你需要填写的值。



T7

考虑一个我们将执行五次连续访问的内存。下表显示了每次访问的类型（读取（加载）、写入（存储））以及访问完成时MAR和MDR的内容。注意，我们已将可寻址性缩短为5位。

Operation No.	R/W	MAR	MDR
1	W	x____	11110
2	-	x____	_____
3	W	x____	10____
4	-	x____	_____
5	-	x____	_____

### Operations on Memory

下表显示了在第一次访问之前、第三次访问之后和第五次访问之后，内存地址x4000到x4004的内容。我们为这个问题添加了一个约束条件以获得一个正确的答案：MDR只能作为加载（读取）访问的结果从内存加载。

Address	Before Access 1	After Access 3	After Access 5
x4000	01101	__0	___
x4001	11010	_0_0	___
x4002	_1__	___	___
x4003	10110	___	01101
x4004	11110	11110	11110

### Contents of Memory locations

You're required to fill in the blanks.

## T8

LC-3没有用于逻辑异或（XOR）功能的操作码。下面的八条指令序列对R1和R2的内容执行异或操作，并将结果放入R3。填写四条缺失的指令，使这八条指令序列能够完成任务。

Address	Instruction
x3000	
x3001	1001 110 010 111111
x3002	0101 101 111 000 010
x3003	
x3004	1001 001 101 111111
x3005	
x3006	
x3007	1001 011 000 111111

## T9

我们希望有一条什么都不做的指令。许多ISA实际上都有一个专门用于什么都不做的操作码。它通常被称为NOP，意思是无操作。该指令被取出、解码和执行。执行阶段就是什么都不做！以下五条指令中哪些可以用作NOP并使程序仍然正常工作？对于其他指令，请描述它们做了什么。

1. 0001 010 001 1 00010
2. 0000 111 000000000
3. 0000 101 000000100
4. 1001 010 111 111111
5. 1111 0000 00100011

## T10

请描述LC-3中BR指令的限制以及JMP指令如何解决这个问题（请给出一个简单的例子）